An Enhanced Levenberg–Marquardt Method via Gram Reduction

Chengchang Liu[1], **Luo Luo**[*2,3], **John C.S. Lui**[1]

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2]School of Data Science, Fudan University
[3]Shanghai Key Laboratory for Contemporary Applied Mathematics
7liuchengchang@gmail.com, luoluo@fudan.edu.cn, cslui@cse.cuhk.edu.hk

Abstract

This paper studies the problem of solving the system of nonlinear equations. We propose the Gram-reduced Levenberg–Marquardt method, which reuses the Gram matrix. Our method has a global convergence guarantee without relying on any step of line-search or solving sub-problems. We show that our method takes a smaller computational complexity than existing Levenberg–Marquardt methods to find the stationary point of the square norm of the equations. We also show that the proposed method enjoys a local superlinear convergence rate under the non-degenerate assumption. Experiments are conducted on real-world applications in scientific computing and machine learning, which validate the efficiency of our method.

1 Introduction

We consider solving the system of nonlinear equations

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{F}(\mathbf{x}) \triangleq [F_1(\mathbf{x}), \ldots, F_d(\mathbf{x})]^\top : \mathbb{R}^d \to \mathbb{R}^d$ with differentiable $F_i(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ for all $i \in [d]$. This problem can also be reformulated by the nonlinear least-square problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \phi(\mathbf{x}) \triangleq \frac{1}{2}\|\mathbf{F}(\mathbf{x})\|^2. \tag{2}$$

Solving nonlinear equations is one of the most fundamental problems in scientific computing (Nesterov 2007; Yuan 2011). It has wide applications in machine learning (Défossez and Bach 2015; Botev, Ritter, and Barber 2017; Bai, Kolter, and Koltun 2019; Liu, Zhu, and Belkin 2022), control system (Berthier, Carpentier, and Bach 2021), data assimilation (Trémolet 2007), and game theory (Frehse and Bensoussan 1984; Nourian and Caines 2013).

First-order methods are popular to solve nonlinear equations. Specifically, one can perform the gradient descent on problem (2), that is

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\nabla\phi(\mathbf{x}_t) = \mathbf{x}_t - \eta\mathbf{J}(\mathbf{x}_t)^\top\mathbf{F}(\mathbf{x}_t), \tag{3}$$

where $\eta > 0$ is the step-size, $\nabla\phi(\mathbf{x}_t) = \mathbf{J}(\mathbf{x}_t)^\top\mathbf{F}(\mathbf{x}_t)$ and $\mathbf{J}(\cdot) \triangleq [\nabla F_1(\cdot), \ldots, \nabla F_d(\cdot)]^\top \in \mathbb{R}^{d\times d}$ is the Jacobian of $\mathbf{F}(\cdot)$. The iteration scheme (3) takes $\mathcal{O}(d^2)$ flops in general. Taking the Lipschitz continuity on $\nabla\phi(\cdot)$, it requires $\mathcal{O}(\epsilon^{-2})$ iterations to find an ϵ-stationary point of $\phi(\cdot)$, then the overall computation cost is $\mathcal{O}(d^2\epsilon^{-2})$. Gauss–Newton methods (Ben-Israel 1966; Nocedal and Wright 1999) consider the estimation $\nabla^2\phi(\cdot) \approx \mathbf{J}(\cdot)^\top\mathbf{J}(\cdot)$ and establish the iteration

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \big(\mathbf{J}(\mathbf{x}_t)^\top\mathbf{J}(\mathbf{x}_t)\big)^\dagger\mathbf{J}(\mathbf{x}_t)^\top\mathbf{F}(\mathbf{x}_t)$$

with local superlinear convergence, while such needs the flops of $\mathcal{O}(d^3)$ to compute the matrix (pseudo) inverse. Recently, Liu and Luo (2022) introduced Broyden family updates (Rodomanov and Nesterov 2021; Lin, Ye, and Zhang 2022) to estimate $\mathbf{J}(\cdot)^\top\mathbf{J}(\cdot)$, which leads to Quasi-Newton methods with local superlinear convergence and reduces the computation cost in per iteration to $\mathcal{O}(d^2)$ flops. However, both Gauss–Newton method and its quasi-Newton variants lack global convergence guarantees.

In this paper, we are interested in Levenberg–Marquardt (LM) methods (Levenberg 1944; Marquardt 1963), which iterates according to

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \big(\mathbf{J}(\mathbf{x}_t)^\top\mathbf{J}(\mathbf{x}_t) + \lambda_t\mathbf{I}\big)^{-1}\mathbf{J}(\mathbf{x}_t)^\top\mathbf{F}(\mathbf{x}_t), \tag{4}$$

where $\lambda_t > 0$ is the regularization parameter. This method globalizes the Gauss–Newton iteration and also maintains local superlinear convergence by properly choosing the regularization term λ_t (Yamashita and Fukushima 2001; Fan and Yuan 2005). However, the iteration scheme (4) takes $\mathcal{O}(d^3)$ flops since it contains matrix inversion such as the Gauss–Newton update. Moreover, the upper bound of the iteration numbers for the existing global convergent LM methods is not better than $\mathcal{O}(\epsilon^{-2})$ (Ueda and Yamashita 2010; Zhao and Fan 2016; Huang and Fan 2018; Bergou, Diouane, and Kungurtsev 2020; Tran-Dinh, Pham, and Nguyen 2020), which makes the total computation cost $\mathcal{O}(d^3\epsilon^{-2})$. In addition, these methods require the line search step or solving the trust region subproblem (Yuan 1994) to achieve a descent direction, which makes their implementation complicated. Recently, Mishchenko (2023) proposed an adaptive LM method without any line-search step nor sub-problem solver, which determine the regularization term by

$$\lambda_t \propto \sqrt{\|\mathbf{J}(\mathbf{x}_t)^\top\mathbf{F}(\mathbf{x}_t)\|}\,. \tag{5}$$

*Corresponding author.

This method can find an ϵ-stationary within at most $\tilde{\mathcal{O}}(\epsilon^{-2.5})$ iterations under the cubic growth condition, which is worse than line-search or trust-region based LM methods. Furthermore, its overall computation cost $\mathcal{O}(d^3\epsilon^{-2.5})$ is worse than the classical LM methods, and the existence of its local superlinear convergence is unknown. Building upon this, there arises the natural question that:

Can we design a computation efficient and globally convergent LM method with local superlinear convergence?

We give an affirmative answer to the above question by proposing the Gram-Reduced Levenberg–Marquardt (GRLM) method. Instead of existing LM methods that compute the Gram matrix $\mathbf{J}(\mathbf{x}_t)^\top \mathbf{J}(\mathbf{x}_t)$ at every iteration, our method only updates $\mathbf{J}(\cdot)^\top \mathbf{J}(\cdot)$ at the snapshot point and reuses it in the next m iterations. The update of our method can be formulated by

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \Big(\mathbf{J}\big(\mathbf{x}_{\pi(t)}\big)^\top \mathbf{J}\big(\mathbf{x}_{\pi(t)}\big) + \lambda_t \mathbf{I}\Big)^{-1} \mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t), \quad (6)$$

where $\pi(t) \triangleq m\lfloor t/m \rfloor$ and λ_t is chosen according to equation (5). Note that the scheme (6) only needs to compute the matrix $\mathbf{J}(\mathbf{x}_{\pi(t)})^\top \mathbf{J}(\mathbf{x}_{\pi(t)})$ in the case of $t \equiv 0 \pmod m$, which significantly reduces the time required to compute the Gram matrix and leads to a better algorithmic complexity. For global behavior, our method takes the overall computational cost of $\mathcal{O}(d^3\epsilon^{-1} + d^2\epsilon^{-2})$ to find an ϵ-stationary point under the cubic-growth condition, which improves the results of $\mathcal{O}(d^3\epsilon^{-2})$ of existing LM methods. For local behavior, our method has the superlinear rate when the solution has a non-degenerate Jacobian, which goes beyond the theory of Mishchenko (2023). We summarize the main theoretical results of GRLM and related work in Table 1.

Paper Organization The remainder of the paper is organized as follows. In Sections 2 and 3, we introduce the preliminaries and related work, respectively. In Section 4, we provide details of the algorithms and convergence analysis of GRLM. In Section 5, we validate our methods by numerical experiments. We conclude our work and discuss the future directions in Section 6. [1]

2 Preliminaries

We use the notation $\|\cdot\|$ to present the Euclidean norm of a vector and the spectral norm of a matrix, respectively. We use the notation $\mathbf{I}$ to present the identity matrix. We denote $\sigma_{\min}(\cdot)$ as the smallest singular value of a matrix. For the ease of presentation, we denote the Gram matrix of $\mathbf{F}(\cdot)$ by

$$\mathbf{G}(\mathbf{x}) \triangleq \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) \succeq \mathbf{0}, \quad (7)$$

where $\mathbf{J}(\mathbf{x}) \in \mathbb{R}^{d\times d}$ is the Jacobian of $\mathbf{F} : \mathbb{R}^d \to \mathbb{R}^d$ at point $\mathbf{x} \in \mathbb{R}^d$. We also define the approximate stationary point of the function $\phi(\cdot) = \frac{1}{2}\|\mathbf{F}(\cdot)\|^2$ as follows.

Definition 1. We say $\mathbf{x} \in \mathbb{R}^d$ is an ϵ-stationary point of $\phi(\cdot)$ if it satisfies $\|\mathbf{J}(\mathbf{x})^\top \mathbf{F}(\mathbf{x})\| \le \epsilon$.

We make the following standard assumptions on the Jacobian of $\mathbf{F}(\cdot)$.

[1]The proof of this paper can be found in https://arxiv.org/pdf/2412.08561.

Assumption 2. We assume the Jacobian of $\mathbf{F}(\cdot)$ is bounded and Lipschitz continuous, that is, we have $\|\mathbf{J}(\mathbf{x})\| \le L_1$ and

$$\|\mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{y})\| \le L_2\|\mathbf{x} - \mathbf{y}\| \quad (8)$$

hold for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and some constants $L_1, L_2 \ge 0$.

Proposition 3. *If the Jacobian $\mathbf{J}(\cdot)$ satisfies Assumption 2, then it holds that*

$$\begin{aligned} &\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})\| \le L_1\|\mathbf{x} - \mathbf{y}\| \quad \textit{and} \\ &\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y}) - \mathbf{J}(\mathbf{y})(\mathbf{x} - \mathbf{y})\| \le \frac{L_2}{2}\|\mathbf{x} - \mathbf{y}\|^2 \end{aligned} \quad (9)$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

The following proposition means that Assumption 2 leads to the Lipschitz continuity of $\mathbf{G}(\cdot)$.

Proposition 4. *If the function $\mathbf{F}(\cdot)$ satisfies Assumption 2, we have*

$$\|\mathbf{G}(\mathbf{x})\| \le L_1^2 \ \textit{and} \ \|\mathbf{G}(\mathbf{y}) - \mathbf{G}(\mathbf{x})\| \le 2L_1L_2\|\mathbf{x} - \mathbf{y}\| \quad (10)$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

3 Related Work

The idea of reusing second-order information dates back to the 1960s, where Shamanskii (1967) presented a variant of the Newton method that constructs a new Jacobian every m iterations and analyzed its local behavior. Later, such an idea has been generalized to different types of Newton methods (Fan 2013; Wang, Chen, and Du 2006; Lampariello and Sciandrone 2001; Adler, Hu, and Lin 2020) and to training the large language models (Liu et al. 2023c; Elbakary et al. 2024). Particularly, Fan (2013) proposed Shamanskii Levenberg–Marquardt method in the form of

$$\begin{aligned} &\mathbf{x}_{t+1} \\ &= \mathbf{x}_t - \Big(\mathbf{J}\big(\mathbf{x}_{\pi(t)}\big)^\top \mathbf{J}\big(\mathbf{x}_{\pi(t)}\big) + \mu_t \|\mathbf{F}(\mathbf{x}_{\pi(t)})\|^\alpha \mathbf{I}\Big)^{-1} \mathbf{J}\big(\mathbf{x}_{\pi(t)}\big)^\top \mathbf{F}(\mathbf{x}_t), \end{aligned}$$

where $\mu_t > 0$ and $\alpha \in [1, 2]$, with a fast local superlinear rate. However, these methods do not have global theoretical advantage compared to the traditional methods.

Recently, Doikov, Chayti, and Jaggi (2023) proposed the regularized Newton method with lazy Hessians for general minimization problem $\min_{\mathbf{x}\in\mathbb{R}^d} \phi(\mathbf{x})$, which iterates with

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \big(\nabla^2\phi(\mathbf{x}_{\pi(t)}) + \sqrt{c\|\nabla\phi(\mathbf{x}_t)\|}\,\mathbf{I}\big)^{-1}\nabla\phi(\mathbf{x}_t) \quad (11)$$

for some $c > 0$. However, this method is not suitable to solve our system of nonlinear equations (or its nonlinear least-square formulation) in the following aspects:

- The update (11) requires accessing the second-order information of the objective. In the view of nonlinear least-square formulation (2), we have

$$\begin{aligned} &\nabla^2\phi(\mathbf{x}_{\pi(t)}) \\ &\quad = \mathbf{J}(\mathbf{x}_{\pi(t)})^\top \mathbf{J}(\mathbf{x}_{\pi(t)}) + \sum_{i=1}^d F_i(\mathbf{x}_{\pi(t)})\nabla^2 F_i(\mathbf{x}_{\pi(t)}). \end{aligned}$$

  Compared with the cost of LM methods mainly depends on $\mathbf{J}(\cdot)$, accessing the Hessians $\nabla^2 F_i(\cdot), \ldots, \nabla^2 F_d(\cdot)$ in above equation may be much more expensive.

Method	Globally Computation Cost	Local superlinear	Line-search Free	References
Gradient Descent	$\mathcal{O}(d^2\epsilon^{-2})$	✗	✓	Nesterov (2018)
Levernberg–Marquardt (v1)	$\mathcal{O}(d^3\epsilon^{-2}\log(\epsilon^{-1}))$	✓	✗	Bergou, Diouane, and Kungurtsev (2020)
Levernberg–Marquardt (v2)	$\mathcal{O}(d^3\epsilon^{-2.5}\log(\epsilon^{-1}))^\dagger$	✗‡	✓	Mishchenko (2023)
GRLM (ours)	$\mathcal{O}(d^2\epsilon^{-2}+d^3\epsilon^{-1})$	✓	✓	Algorithm 1

† This results can be improved to $\mathcal{O}(d^3\epsilon^{-2.5})$ by using our analysis framework. We present a discussion in Remark 10.

‡ Our analysis can also provide the local superlinear convergence rate for Levernberg–Marquardt (v2), which is not proved in Mishchenko (2023). We present the result in Remark 16.

Table 1: We summarize the globally convergent algorithms for solving the nonlinear equations by their total computation cost for finding the ϵ-stationary point of $\phi(\cdot)$ (**Globally Computation Cost**), if they enjoy local superlinear rates (**Local Superlinear**), and if they are line-search free (**Line-search Free**).

Algorithm 1: Gram-Reduced Levenberg–Marquardt (GRLM)

Input: $\mathbf{x}_0$, c, T, and m
for $t = 0, 1, \dots T-1$
  $\pi(t) = m\lfloor t/m \rfloor, \quad \mathbf{z}_t = \mathbf{x}_{\pi(t)}$
  $\lambda_t = \sqrt{c\|\mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t)\|}$
  $\mathbf{x}_{t+1} = \mathbf{x}_t - \big(\mathbf{G}(\mathbf{z}_t) + \lambda_t \mathbf{I}\big)^{-1}\mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t)$
end for

- The convergence guarantees of (11) require the convexity of $\phi(\cdot)$ and the Lipschitz continuity of $\nabla^2\phi(\cdot)$, while the function $\phi(\cdot) = \frac{1}{2}\|\mathbf{F}(\cdot)\|_2$ in our problem (2) is non-convex in general and the popular setting of nonlinear equations (Assumption 2) cannot guarantee the Lipschitz continuity on $\nabla\phi(\cdot)$ nor $\nabla^2\phi(\cdot)$.
- It is also notable that Doikov, Chayti, and Jaggi (2023) also studied the cubic regularized Newton methods (Nesterov 2006) with lazy Hessians for the general non-convex case, however, the algorithm also requires accessing the Hessian $\nabla^2\phi(\cdot)$ and its analysis is based on the Lipschitz continuity of Hessian. Thus, their theory is not applicable to solving the system of nonlinear equations.

4 The Gram-Reduced LM Method

We propose the Gram-Reduced Levenberg–Marquardt (GRLM) method in Algorithm 1. Our GRLM method takes the Gram matrix $\mathbf{G}(\mathbf{z}_t) = \mathbf{J}(\mathbf{z}_t)^\top \mathbf{J}(\mathbf{z}_t)$ as the approximation to $\nabla^2\phi(\mathbf{z}_t)$, which avoids the exact $\nabla^2\phi(\cdot)$ in regularized Newton with Lazy Hessians (Doikov, Chayti, and Jaggi 2023). Furthermore, setting $\mathbf{z}_t$ in GRLM does not require accessing $\mathbf{J}(\cdot)^\top\mathbf{J}(\cdot)$ in all iterations, making the algorithm more efficient than existing LM methods.

We then consider the convergence of our GRLM method. For the ease of presentation, we denote

$$\lambda_t \triangleq \sqrt{c\|\mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t)\|} \quad \text{and} \quad r_t \triangleq \|\mathbf{x}_{t+1} - \mathbf{x}_t\|.$$

where the constant $c > 0$ follows the input of Algorithm 1.

In contrast to the analysis of Mishchenko (2023) which only lower bounds r_t by λ_{t+1}, the following lemma provides both the lower bound and the upper bound of r_t by λ_t.

Lemma 5. *Under Assumption 2, Algorithm 1 holds that*

$$\frac{\lambda_t^2}{c(L_1^2 + \lambda_t)} \le r_t \le \frac{\lambda_t}{c}. \tag{12}$$

Compared to the ordinary LM method, our GRLM use $\mathbf{G}(\mathbf{z}_t)$ instead of $\mathbf{G}(\mathbf{x}_t)$ at each iteration, which leads to some additional error terms. In the following lemma, we show that such terms can be bounded by the distance between the current iteration point to the snapshot point due to the Lipschitz continuity of $\mathbf{G}(\cdot)$ we have proved in Proposition 4.

Lemma 6. *Under Assumption 2, Algorithm 1 holds that*

$$\begin{aligned}&\big\|\mathbf{G}(\mathbf{x}_t)(\mathbf{x}_{t+1} - \mathbf{x}_t) + \mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t)\big\| \\ &\quad \le \lambda_t r_t + 2L_1 L_2 r_t \|\mathbf{z}_t - \mathbf{x}_t\|,\end{aligned}$$

and

$$\begin{aligned}&\big\langle \mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x}_t)(\mathbf{x}_{t+1} - \mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t\big\rangle \\ &\quad \le -\lambda_t r_t^2 + 2L_1 L_2 r_t^2 \|\mathbf{x}_t - \mathbf{z}_t\|.\end{aligned} \tag{13}$$

4.1 Global Convergence Analysis

We establish the global convergence for Algorithm 1 in this subsection. We adopt the following cubic-growth assumption which has been well studied by Mishchenko (2023).

Assumption 7. We assume the function $\mathbf{F} : \mathbb{R}^d \to \mathbb{R}^d$ satisfies that

$$\|\mathbf{F}(\mathbf{y})\|^2 \le \|\mathbf{F}(\mathbf{x}) + \mathbf{J}(\mathbf{x})(\mathbf{y} - \mathbf{x})\|^2 + M\|\mathbf{y} - \mathbf{x}\|^3 \tag{14}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, where $M > 0$ is some constant.

The following lemma shows that, by properly setting the regularization parameter $c > 0$, we can guarantee the descent property of $\|\mathbf{F}(\mathbf{x}_t)\|$ at the snapshot point $\mathbf{x}_t \in \mathbb{R}^d$ such that $t \equiv 0 \pmod m$.

Lemma 8. *Under Assumption 2 and 7, if we run Algorithm 1 with $c \triangleq \max\{4L_1L_2m, M\}$, then it holds that*

$$\|\mathbf{F}(\mathbf{x}_{(k+1)m})\|^2 \le \|\mathbf{F}(\mathbf{x}_{km})\|^2 \le \cdots \|\mathbf{F}(\mathbf{x}_0)\|^2$$

and

$$\|\mathbf{F}(\mathbf{x}_{km})\|^2 - \|\mathbf{F}(\mathbf{x}_{(k+1)m})\|^2 \geq \sum_{t=km}^{(k+1)m-1} \frac{r_t^2 \lambda_t}{6},$$

for all $k = 0, 1, \cdots$.

Now, we present the global convergence results of our GRLM (Algorithm 1).

Theorem 9. *Following the settings of Lemma 8, Algorithm 1 takes at most*

$$T = \mathcal{O}(c^2 + c^{-0.5}\epsilon^{-2.5})$$

iterations to find an ϵ*-stationary point of* $\phi(\cdot)$*, i.e., we have*

$$\min_{t=0,\cdots,T-1} \|\mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t)\| \leq \epsilon.$$

Remark 10. Taking $m = 1$ such that $c = \max\{4L_1L_2, M\}$, Algorithm 1 reduces to the ordinary Levernberg–Mardquardt method proposed by Mishchenko (2023). Compared with the iteration complexity of $\mathcal{O}(\epsilon^{-2.5}\log(1/\epsilon))$ established in the previous work, Theorem 1 provides an improved iteration complexity of $\mathcal{O}(\epsilon^{-2.5})$, which removes the logarithmic factor. We achieve this by lower bounding r_t by λ_t rather than λ_{t+1} in the existing work and using a novel proof framework that divides the iterations according to the value of λ_t.

Recall that Algorithm 1 needs to compute a new $\mathbf{G}(\mathbf{z}_t)$ in every m iterations. W.L.O.G, we let $m \geq \frac{M}{4L_1L_2}$ such that $c = 4L_1L_2m$ and use K to denote the numbers of the snapshot points, then Theorem 9 means

$$K = \left\lceil \frac{T}{m} \right\rceil = \mathcal{O}(m + m^{-1.5}\epsilon^{-2.5}).$$

For reusing the Gram matrix $\mathbf{G}(\mathbf{z}_t)$ and reducing the computation cost, we can implement Algorithm 1 by performing singular value decomposition on the Jacobian at the snapshot point, which generally takes $\mathcal{O}(d^3)$ flops and results

$$\mathbf{J}(\mathbf{z}_t) = \mathbf{U}(\mathbf{z}_t)\mathbf{\Sigma}(\mathbf{z}_t)\mathbf{V}(\mathbf{z}_t)^\top,$$

where $\mathbf{U}(\mathbf{z}_t), \mathbf{V}(\mathbf{z}_t) \in \mathbb{R}^{d\times d}$ are orthogonal and $\mathbf{\Sigma}(\mathbf{z}_t)$ is diagonal. Based on the SVD of $\mathbf{J}(\mathbf{z}_t)$, we can update $\mathbf{x}_t$ according to

$$\begin{aligned}\mathbf{x}_{t+1} &= \mathbf{x}_t - \left(\mathbf{G}(\mathbf{z}_t) + \lambda_t \mathbf{I}\right)^{-1} \mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t) \\ &= \mathbf{x}_t - \mathbf{V}(\mathbf{z}_t)\big((\mathbf{\Sigma}(\mathbf{z}_t))^2 + \lambda_t \mathbf{I}\big)^{-1} \mathbf{V}(\mathbf{z}_t)^\top \mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t),\end{aligned}$$

which can be done in $\mathcal{O}(d^2)$ flops.

Thus, the total computation cost of Algorithm 1 is

$$\begin{aligned}&\#\text{flops} = \mathcal{O}(d^3K + d^2T) \\ &= \mathcal{O}(d^3m + d^3m^{-1.5}\epsilon^{-2.5} + d^2m^2 + d^2m^{-0.5}\epsilon^{-2.5}).\end{aligned} \tag{15}$$

Now we desire to select an appropriate m to minimize the overall flops. We let $m = \mathcal{O}(d^\alpha \epsilon^{-\beta})$ for some $\alpha, \beta \geq 0$ and plug it into equation (15), then the total computation cost is

$$\begin{aligned}&\#\text{flops} \\ &= \mathcal{O}\big(d^{3+\alpha}\epsilon^{-\beta} + d^{3-1.5\alpha}\epsilon^{-2.5+1.5\beta} \\ &\quad + d^{2+2\alpha}\epsilon^{-2\beta} + d^{2-0.5\alpha}\epsilon^{-2.5+0.5\beta}\big) \\ &\geq \mathcal{O}\Big(d^{3-0.25\alpha}\epsilon^{-1.25+0.25\beta} + d^{2+1.25\alpha}\epsilon^{-1.25-0.75\beta}\Big),\end{aligned}$$

where the inequality is due to the AM–GM and the equality holds when $\alpha = 0$ and $\beta = 1$. Therefore, we should take $m = \Theta(\epsilon^{-1})$, which leads to $\#\text{flops} = \mathcal{O}(d^3\epsilon^{-1} + d^2\epsilon^{-2})$. We formally present the above result in the following corollary.

Corollary 11. *Following the setting of Lemma 8, running Algorithm 1 with* $m = \Theta(\epsilon^{-1})$ *can find an* ϵ*-stationary point of* $\phi(\cdot)$ *within* $\mathcal{O}(d^3\epsilon^{-1} + d^2\epsilon^{-2})$ *flops.*

As comparison, the globally convergent LM method proposed by Mishchenko (2023) requires $\mathcal{O}(d^3\epsilon^{-2.5}\log(1/\epsilon))$ flops to find an ϵ-stationary point of $\phi(\cdot)$, which is more expensive than our complexity of $\mathcal{O}(d^3\epsilon^{-1} + d^2\epsilon^{-2})$. Our result is also sharper than the complexity of $\mathcal{O}(d^3\epsilon^{-2})$ for other LM methods (Ueda and Yamashita 2010; Zhao and Fan 2016; Huang and Fan 2018; Bergou, Diouane, and Kungurtsev 2020; Tran-Dinh, Pham, and Nguyen 2020).

4.2 Local Convergence Analysis

We establish the local superlinear rates for Algorithm 1 in this subsection, with the following assumption for our problem.

Assumption 12. We assume that there exists a solution $\mathbf{x}^*$ for problem (1) which has non-degenerate Jacobian, i.e., we have

$$\mathbf{F}(\mathbf{x}^*) = \mathbf{0} \quad \text{and} \quad \sigma_{\min}(\mathbf{J}(\mathbf{x}^*)) = \mu \tag{16}$$

for some $\mu > 0$.

The following proposition says that Assumption 12 guarantees that the points in the local neighbor of solution $\mathbf{x}^*$ have nondegenerate $\mathbf{J}(\cdot)$ and $\mathbf{G}(\cdot)$.

Proposition 13 ((Liu et al. 2023a, Proposition 2.3))**.** *Under Assumption 2 and 12, for all* $\mathbf{x} \in \mathbb{R}^d$ *satisfying* $\|\mathbf{x} - \mathbf{x}^*\| \leq \mu^2/(6L_1L_2)$*, we have* $\sigma_{\min}(\mathbf{J}(\mathbf{x})) \geq \frac{\mu}{\sqrt{2}}$ *and* $\mathbf{G}(\mathbf{x}) \succeq \frac{\mu^2}{2}\mathbf{I}$.

We first provide the recursion for the distances between the points in the local region to the solution.

Lemma 14. *Under Assumptions 2 and 12, we assume that some points* $\mathbf{x}_t$ *and* $\mathbf{z}_t$ *generated by Algorithm 1 are sufficiently close to the solution* $\mathbf{x}^*$ *such that*

$$\begin{aligned}&\mathbf{x}_t, \mathbf{z}_t \in \mathcal{S} \\ &\triangleq \Big\{\mathbf{x} : \mathbf{x} \in \mathbb{R}^d, \|\mathbf{x} - \mathbf{x}^*\| \leq \min\Big\{\frac{\mu^2}{18L_1L_2}, \frac{\mu^4}{64L_1^2c}\Big\}\Big\},\end{aligned} \tag{17}$$

then it holds that

$$\begin{aligned}\|\mathbf{x}_{t+1} - \mathbf{x}^*\| \leq &\alpha_1\|\mathbf{x}_t - \mathbf{x}^*\|^2 + \alpha_2\|\mathbf{x}_t - \mathbf{x}^*\|^{1.5} \\ &+ 2\alpha_1\|\mathbf{z}_t - \mathbf{x}^*\|\|\mathbf{x}_t - \mathbf{x}^*\|,\end{aligned} \tag{18}$$

where $\alpha_1 \triangleq L_1L_2/\mu^2$ *and* $\alpha_2 \triangleq 2L_1c^{0.5}/\mu^2$*. In addition, we have* $\mathbf{x}_{t+1} \in \mathcal{S}$.

Now, we show the explicit local superlinear convergence rate of GRLM.

Theorem 15. *Under Assumptions 2 and 12, we run Algorithm 1 with* $\mathbf{x}_0$ *such that*

$$\|\mathbf{x}_0 - \mathbf{x}^*\| \leq \frac{1}{32(\alpha_1 + \alpha_2^2)}, \tag{19}$$

where α_1 *and* α_2 *follow the definitions in Lemma 14. Then it holds that*

$$\|\mathbf{x}_t - \mathbf{x}^*\| \leq \frac{1}{2(\alpha_1 + \alpha_2^2)} \left(\frac{1}{2}\right)^{2(1+(1+m/2)^{\pi(t)})(1+(t\%m)/2)},$$

where $t\%m = t - \pi(t)$.

Remark 16. When $m = 1$, Theorem 15 provides the superlinear rate of $\|\mathbf{x}_t - \mathbf{x}^*\| \leq \frac{1}{2(\alpha_1+\alpha_2^2)} \cdot \left(\frac{1}{2}\right)^{4(1+t/2)}$, for the LM method in Mishchenko (2023, Algorithm 2.2).

The local convergence behavior has been widely studied for LM methods, while most of the work focuses on selecting the regularization parameter in the order of $\lambda_t \propto \|\mathbf{F}(\mathbf{x}_t)\|^\alpha$ (Yamashita and Fukushima 2001; Fan and Yuan 2005; Bergou, Diouane, and Kungurtsev 2020). This paper first investigates local convergence using $\lambda_t \propto \sqrt{\|\mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t)\|}$.

The local superlinear rate in Theorem 15 is comparable to the rate achieved by regularized Newton with lazy Hessians (Doikov, Chayti, and Jaggi 2023). Compared with Doikov, Chayti, and Jaggi (2023) where the objective is supposed to be strongly convex, we only assume the Jacobian at the solution is nondegenerate, which allows the objective $\phi(\cdot) = \|\mathbf{F}(\cdot)\|^2$ to be non-convex. Furthermore, we characterize the superlinear convergence of GRLM by the measure of distance to the solution, while the analysis of Doikov, Chayti, and Jaggi (2023) considers the gradient norm of the objective.

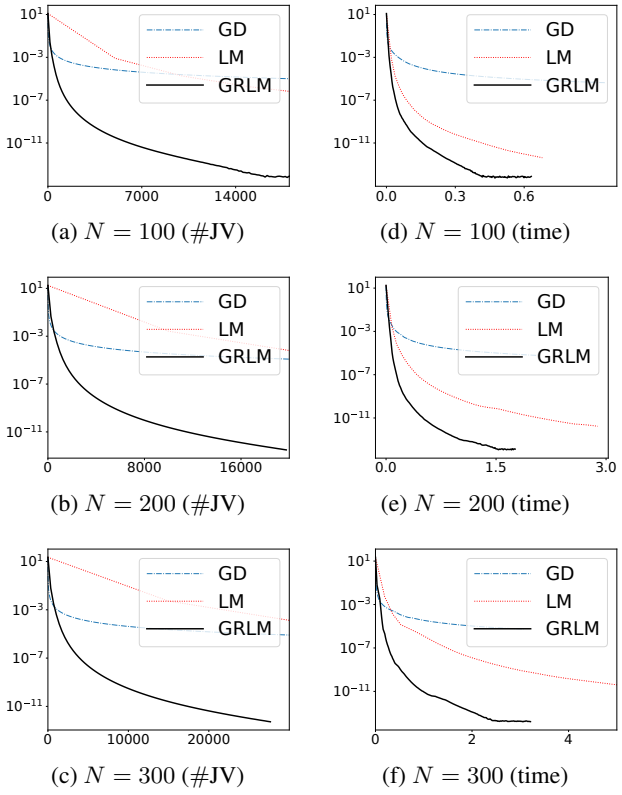
The GRLM method (Algorithm 1) takes an average computation cost of $\mathcal{O}(d^2)$ per iteration when we choose $m = \Omega(d)$, which matches the cost of per iteration in quasi-Newton methods or incremental Newton methods to solve the nonlinear equations (Lin, Ye, and Zhang 2021; Ye, Lin, and Zhang 2021; Liu and Luo 2022; Liu et al. 2023a; Zhou et al. 2024). However, these quasi-Newton methods lack global convergence guarantees.

5 Experiment

In this section, we conduct experiments on scientific computing and machine learning applications of Chandrasekhar H-equation and non-convex regularized logistic regression to verify our theory.

We choose the gradient descent method (GD) and the Levernberg–Marquardt method with gradient regularization (Mishchenko 2023) (LM) as baselines. We do not compare our methods with quasi-Newton type methods (Ye, Lin, and Zhang 2021; Lin, Ye, and Zhang 2021; Liu et al. 2023a) nor Jacobian-free Newton–Krylov methods (Knoll and Keyes 2004; Ashrafizadeh, Devaud, and Aydemir 2015), since they do not have global convergence guarantees.

For all experiments, we tuned the step size η for GD from $\{0.1, 0.2, \cdots, 1\}$. We tune the regularized parameter c in LM and GRLM from $\{1, 10, 100, 1000\}$. Our experiments are conducted on a PC with Apple M1 and all algorithms are implemented in Python 3.8.12.



(a) $N = 100$ (#JV) (d) $N = 100$ (time)

(b) $N = 200$ (#JV) (e) $N = 200$ (time)

(c) $N = 300$ (#JV) (f) $N = 300$ (time)

Figure 1: We demonstrate Jacobian-vector products computing times (#JV) and CPU time (second) vs. $\|\mathbf{J}(\mathbf{x})^\top \mathbf{F}(\mathbf{x})\|_2$ for H-equation with different equation numbers N.

5.1 Chandrasekhar H-equation

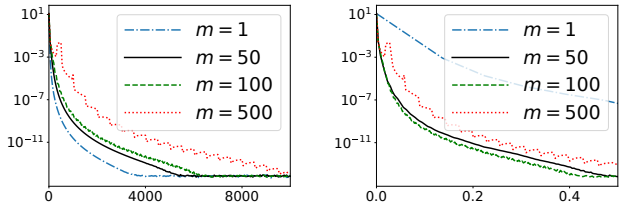
Chandrasekhar H-equation plays an important role in scientific computing (Chandrasekhar 1960; Leggett 1976; Kelley 1982) and has been well studied in the previous literature (Kelley 1995; Lin, Ye, and Zhang 2021; Ye, Lin, and Zhang 2021; Liu et al. 2023a). It is defined by

$$F_i(\mathbf{x}) \triangleq x_i - \left(1 - \frac{c}{2N} \sum_{j=1}^{N} \frac{\mu_i x_j}{\mu_i + \mu_j}\right)^{-1},$$

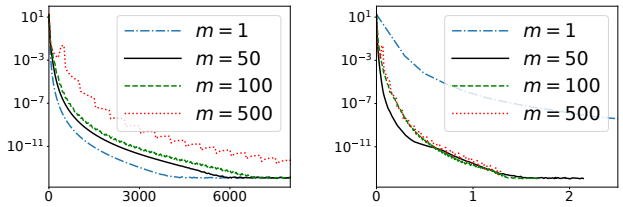
where $\mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), \cdots, F_N(\mathbf{x})]^\top \in \mathbb{R}^N$ and $\mathbf{x} = [x_1, \cdots, x_N]^\top \in \mathbb{R}^N$.

We compare GRLM ($m = 50$) with baselines. We test the cases $N = 100$, $N = 200$, and $N = 300$. In all cases, we set $c = 1 - 10^{-10}$. We randomize an $\mathbf{x}_0$ as the initial points for all the methods. The results of the total number of the Jacobian-vector products (#JV) computed in the algorithms against $\|\mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t)\|$ and the running time against $\|\mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t)\|$ is presented in Figure 1.[2] We observe that the proposed Gram-reduced Levernberg–Mardquardt method (GRLM) outperforms the baselines in all cases. We also provide experiments to study the impact of choosing different m in GRLM (Algorithm 1). We choose $m = \{1, 50, 100, 500\}$ for all cases and present the results in Figure 2.

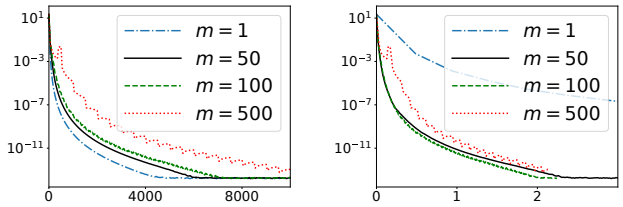
[2]The number of the Jacobian-vector products for computing a full Jacobian is d.



(a) $N = 100$ (iteration) (d) $N = 100$ (time)

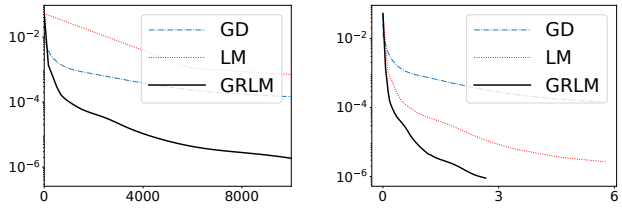


(b) $N = 200$ (iteration) (e) $N = 200$ (time)

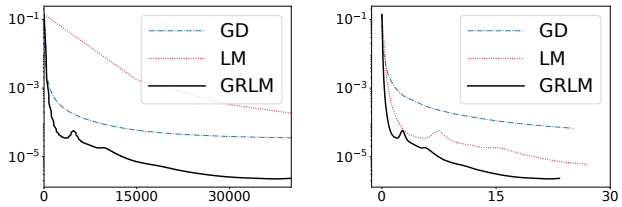


(c) $N = 300$ (iteration) (f) $N = 300$ (time)

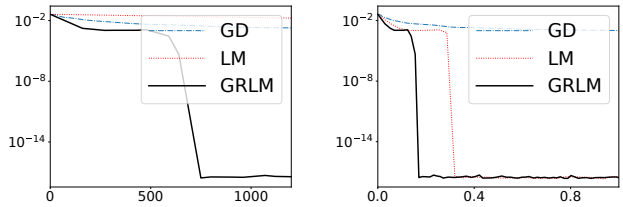
Figure 2: We demonstrate the iteration numbers (iteration) and CPU time (second) vs. $\|\mathbf{J}(\mathbf{x})^\top \mathbf{F}(\mathbf{x})\|_2$ for H-equation with different equation numbers N.



(a) "a1a" (#JV) (d) "a1a" (time)



(b) "w1a" (#JV) (e) "w1a" (time)



(c) "splice" (#JV) (f) "splice" (time)

Figure 3: We demonstrate Jacobian-vector products computing times (#JV) and CPU time (second) vs. $\|\mathbf{J}(\mathbf{x})^\top \mathbf{F}(\mathbf{x})\|_2$ for non-convex logistic regression on datasets "a1a", "w1a", and "splice".

We find that a smaller m leads to a faster iteration in (a), (b), and (c) of Figure 2, which matches our theoretical results obtained in Theorem 9. However, there exists a trade-off between the iteration complexity and the computation cost per iteration.

5.2 Non-Convex Regularized Logistic Regression

We further validate the GRLM methods on the non-convex regularized logistic regression model (Antoniadis, Gijbels, and Nikolova 2011):

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n} \ln(1+\exp(-b_i \mathbf{a}_i^\top \mathbf{x})) + \lambda \sum_{p=1}^{d} \frac{x_{(p)}^2}{1+x_{(p)}^2},$$

where $x_{(p)}$ is the p-th coordinate of $\mathbf{x} \in \mathbb{R}^d$, $\lambda > 0$ is the regularized parameter, $\mathbf{a}_i \in \mathbb{R}^d$ and $b_i \in \{-1, +1\}$ are the feature and the corresponding label of the i-th sample. This model corresponds to solving the following nonlinear equations $\mathbf{F}(\mathbf{x}) \triangleq \nabla f(\mathbf{x})$.

We compare GRLM ($m = 100$) with baselines in three real-world datasets: "a1a" ($n = 1,605$, $d = 123$), "w1a" ($n = 2,477$, $d = 300$), and "splice" ($n = 1,000$, $d = 60$). All of these data sets can be downloaded from the LIBSVM repository (Chang and Lin 2011). We present the results of the number of Jacobian-vector products (#JV) against $\|\mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t)\|$ and the running time against $\|\mathbf{J}(\mathbf{x}_t)^\top \mathbf{F}(\mathbf{x}_t)\|$ in Figure 3. GRLM outperforms the baselines for all datasets in terms of both the total number of Jacobian vector products and the CPU time.

6 Conclusion and Future Work

In this paper, we have proposed Gram-Reduced Levenberg–Marquardt Method (GRLM) to solve nonlinear equations. GRLM improves the behavior of existing LM methods by reducing the computation times of Gram matrices. It is globally convergent with a simple iteration scheme and a low computational cost. Furthermore, it exhibits local superlinear convergence, which cannot be achieved by any of the first-order methods. To the best of our knowledge, this is the first method for solving nonlinear equations that can achieve the best of both worlds.

For future work, it is interesting to design sketched, stochastic, and distributed variants of GRLM (Yuan 2009; Yuan, Lazaric, and Gower 2022; Chayti, Doikov, and Jaggi 2023; Liu et al. 2023b). It is also possible to incorporate the idea of super universal Newton methods (Doikov, Mishchenko, and Nesterov 2024) to make GRLM parameter-free.

Acknowledgments

Chengchang Liu is supported by the National Natural Science Foundation of China (624B2125). Luo Luo is supported by the National Natural Science Foundation of China (No. 62206058), Shanghai Sailing Program (22YF1402900), Shanghai Basic Research Program (23JC1401000), and the Major Key Project of PCL under Grant PCL2024A06. John C.S. Lui is supported in part by the RGC GRF-142202923.

References

Adler, I.; Hu, Z. T.; and Lin, T. 2020. New proximal Newton-type methods for convex optimization. In *2020 59th IEEE Conference on Decision and Control (CDC)*, 4828–4835. IEEE.

Antoniadis, A.; Gijbels, I.; and Nikolova, M. 2011. Penalized likelihood regression for generalized linear models with non-quadratic penalties. *Annals of the Institute of Statistical Mathematics*, 63: 585–615.

Ashrafizadeh, A.; Devaud, C.; and Aydemir, N. 2015. A Jacobian-free Newton–Krylov method for thermalhydraulics simulations. *International Journal for Numerical Methods in Fluids*, 77(10): 590–615.

Bai, S.; Kolter, J. Z.; and Koltun, V. 2019. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32.

Ben-Israel, A. 1966. A Newton-Raphson method for the solution of systems of equations. *Journal of Mathematical analysis and applications*, 15(2): 243–252.

Bergou, E. H.; Diouane, Y.; and Kungurtsev, V. 2020. Convergence and complexity analysis of a Levenberg–Marquardt algorithm for inverse problems. *Journal of Optimization Theory and Applications*, 185: 927–944.

Berthier, E.; Carpentier, J.; and Bach, F. 2021. Fast and Robust Stability Region Estimation for Nonlinear Dynamical Systems. In *2021 European Control Conference (ECC)*, 1412–1419. IEEE.

Botev, A.; Ritter, H.; and Barber, D. 2017. Practical Gauss-Newton optimisation for deep learning. In *International Conference on Machine Learning*, 557–565. PMLR.

Chandrasekhar, S. 1960. *Radiative transfer*. Courier Corporation.

Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2: 27:1–27:27. Software and datasets available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chayti, E. M.; Doikov, N.; and Jaggi, M. 2023. Unified Convergence Theory of Stochastic and Variance-Reduced Cubic Newton Methods. *arXiv preprint arXiv:2302.11962*.

Défossez, A.; and Bach, F. 2015. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Artificial Intelligence and Statistics*, 205–213. PMLR.

Doikov, N.; Chayti, E. M.; and Jaggi, M. 2023. Second-order optimization with lazy Hessians. In *International Conference on Machine Learning*, 8138–8161. PMLR.

Doikov, N.; Mishchenko, K.; and Nesterov, Y. 2024. Super-universal regularized Newton method. *SIAM Journal on Optimization*, 34(1): 27–56.

Elbakary, A.; Issaid, C. B.; Shehab, M.; Seddik, K.; ElBatt, T.; and Bennis, M. 2024. Fed-Sophia: A Communication-Efficient Second-Order Federated Learning Algorithm. *arXiv preprint arXiv:2406.06655*.

Fan, J. 2013. A Shamanskii-like Levenberg-Marquardt method for nonlinear equations. *Computational Optimization and Applications*, 56(1): 63–80.

Fan, J.; and Yuan, Y.-X. 2005. On the quadratic convergence of the Levenberg–Marquardt method without nonsingularity assumption. *Computing*, 74: 23–39.

Frehse, J.; and Bensoussan, A. 1984. Nonlinear elliptic systems in stochastic game theory. *Journal für die reine und angewandte Mathematik*, 350: 23–67.

Huang, J.-C.; and Fan, J.-Y. 2018. Global Complexity Bound of the Inexact Levenberg–Marquardt Method. *Journal of the Operations Research Society of China*, 6: 417–428.

Kelley, C. 1982. Approximate methods for the solution of the Chandrasekhar H-equation. *Journal of Mathematical Physics*, 23(11): 2097–2100.

Kelley, C. T. 1995. *Iterative methods for linear and nonlinear equations*. SIAM.

Knoll, D. A.; and Keyes, D. E. 2004. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2): 357–397.

Lampariello, F.; and Sciandrone, M. 2001. Global convergence technique for the Newton method with periodic Hessian evaluation. *Journal of optimization theory and applications*, 111: 341–358.

Leggett, R. W. 1976. A new approach to the H-equation of Chandrasekhar. *SIAM Journal on Mathematical Analysis*, 7(4): 542–550.

Levenberg, K. 1944. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2): 164–168.

Lin, D.; Ye, H.; and Zhang, Z. 2021. Explicit superlinear convergence rates of Broyden's methods in nonlinear equations. *arXiv preprint arXiv:2109.01974*.

Lin, D.; Ye, H.; and Zhang, Z. 2022. Explicit convergence rates of greedy and random quasi-Newton methods. *Journal of Machine Learning Research*, 23(162): 1–40.

Liu, C.; Chen, C.; Luo, L.; and Lui, J. C. 2023a. Block Broyden's Methods for Solving Nonlinear Equations. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Liu, C.; Chen, L.; Luo, L.; and Lui, J. C. 2023b. Communication efficient distributed Newton method with fast convergence rates. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1406–1416.

Liu, C.; and Luo, L. 2022. Quasi-Newton methods for saddle point problems. *Advances in Neural Information Processing Systems*, 35: 3975–3987.

Liu, C.; Zhu, L.; and Belkin, M. 2022. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59: 85–116.

Liu, H.; Li, Z.; Hall, D.; Liang, P.; and Ma, T. 2023c. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*.

Marquardt, D. W. 1963. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2): 431–441.

Mishchenko, K. 2023. Regularized Newton Method with Global $O(1/k^2)$ Convergence. *SIAM Journal on Optimization*, 33(3): 1440–1462.

Nesterov, Y. 2006. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1): 177–205.

Nesterov, Y. 2007. Modified Gauss–Newton scheme with worst case guarantees for global performance. *Optimisation methods and software*, 22(3): 469–483.

Nesterov, Y. 2018. *Lectures on convex optimization*, volume 137. Springer.

Nocedal, J.; and Wright, S. J. 1999. *Numerical optimization*. Springer.

Nourian, M.; and Caines, P. E. 2013. ϵ-Nash mean field game theory for nonlinear stochastic dynamical systems with major and minor agents. *SIAM Journal on Control and Optimization*, 51(4): 3302–3331.

Rodomanov, A.; and Nesterov, Y. 2021. Greedy quasi-Newton methods with explicit superlinear convergence. *SIAM Journal on Optimization*, 31(1): 785–811.

Shamanskii, V. 1967. A modification of Newton's method. *Ukrainian Mathematical Journal*, 19(1): 118–122.

Tran-Dinh, Q.; Pham, N.; and Nguyen, L. 2020. Stochastic Gauss-Newton algorithms for nonconvex compositional optimization. In *International Conference on Machine Learning*, 9572–9582. PMLR.

Trémolet, Y. 2007. Model-error estimation in 4D-Var. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 133(626): 1267–1280.

Ueda, K.; and Yamashita, N. 2010. On a global complexity bound of the Levenberg-Marquardt method. *Journal of optimization theory and applications*, 147: 443–453.

Wang, C.-y.; Chen, Y.-y.; and Du, S.-q. 2006. Further insight into the Shamanskii modification of Newton method. *Applied mathematics and computation*, 180(1): 46–52.

Yamashita, N.; and Fukushima, M. 2001. On the rate of convergence of the Levenberg-Marquardt method. In *Topics in Numerical Analysis: With Special Emphasis on Nonlinear Problems*, 239–249. Springer.

Ye, H.; Lin, D.; and Zhang, Z. 2021. Greedy and Random Broyden's Methods with Explicit Superlinear Convergence Rates in Nonlinear Equations. *arXiv preprint arXiv:2110.08572*.

Yuan, R.; Lazaric, A.; and Gower, R. M. 2022. Sketched Newton–Raphson. *SIAM Journal on Optimization*, 32(3): 1555–1583.

Yuan, Y.-X. 1994. *Trust region algorithms for nonlinear equations*. Citeseer.

Yuan, Y.-X. 2009. Subspace methods for large scale nonlinear equations and nonlinear least squares. *Optimization and Engineering*, 10(2): 207–218.

Yuan, Y.-X. 2011. Recent advances in numerical methods for nonlinear equations and nonlinear least squares. *Numerical algebra, control & optimization*, 1(1): 15.

Zhao, R.; and Fan, J. 2016. Global complexity bound of the Levenberg–Marquardt method. *Optimization Methods and Software*, 31(4): 805–814.

Zhou, Z.; Liu, Z.; Liu, C.; and Luo, L. 2024. Incremental Gauss–Newton Methods with Superlinear Convergence Rates. *arXiv preprint arXiv:2407.03195*.